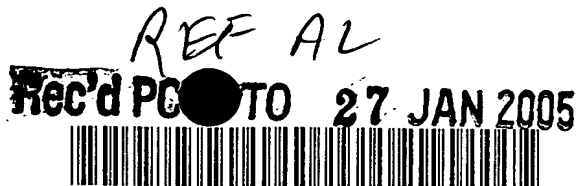


Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11) EP 1 139 275 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:  
04.10.2001 Bulletin 2001/40

(51) Int Cl.7: G06K 15/00, G06F 17/27,  
G07B 17/00

(21) Application number: 01107742.7

(22) Date of filing: 02.04.2001

(84) Designated Contracting States:  
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU  
MC NL PT SE TR  
Designated Extension States:  
AL LT LV MK RO SI

- Clemens, Daniel J.  
Deer Harbor, Washington 98243 (US)
- Girardi, Victor  
Oxford, Connecticut 06478-1839 (US)
- Rolling, Patrick  
Cheshire, Connecticut 06410 (US)

(30) Priority: 31.03.2000 US 541001

(71) Applicant: PITNEY BOWES INC.  
Stamford, Connecticut 06926-0700 (US)

(74) Representative: HOFFMANN - EITLE  
Patent- und Rechtsanwälte  
Arabellastrasse 4  
81925 München (DE)

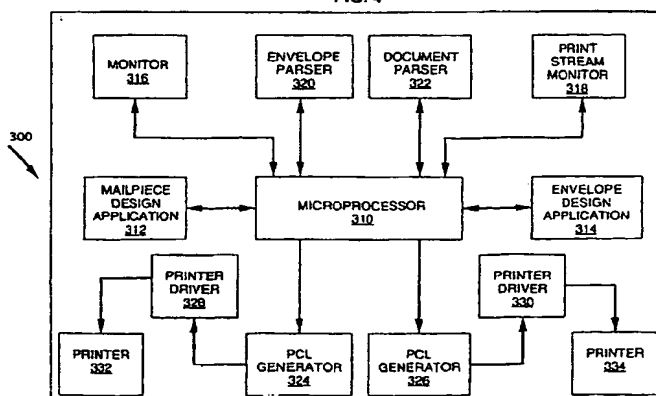
(72) Inventors:  
• Bratsamos, James S.  
Seattle, Washington 98122 (US)

(54) **A method and system of print stream address extraction**

(57) The invention is a method and system of modifying print stream data. The method begins by sending a print stream from an application through a graphical device interface to a print spooler. A print stream monitor determines whether or not the print stream comprises a set of text data and/or a set of address data. If the print stream comprises text data, then the text data is tagged and sent to a user mode module where it is stored in a local buffer. The tagged and stored text data is then retrieved and parsed for any address which may be contained within the tagged text. The address data is then extracted from the print stream and placed in an envelope print format to create an envelope data set. An en-

velope printer device context is then created and the envelope data set is transmitted to an envelope kernel for creating an envelope printer device language file. The print stream is converted by a document printer command language (PCL) generator into an envelope printer language. The envelope data set is then printed. The envelope data set is injected back into the print stream from which it was extracted by merging the set of text data and the set of envelope data. The print stream is then transmitted to a next destination such as a document printer, where a printer driver is then activated for causing a printer to print the document data to one or more sheets.

FIG. 4



**Description**

[0001] Reference is made to European Patent Publication Number 0981111, entitled A METHOD AND SYSTEM OF DISPLAYING DATABASE CONTENTS IN ENVELOPE DATA FIELDS, assigned to the assignee of this application and filed on July 20, 1999.

[0002] Reference is made to European Patent Publication Number 0974915, entitled A METHOD AND SYSTEM OF PRINT STREAM ADDRESS EXTRACTION, assigned to the assignee of this application and filed on July 20, 1999.

[0003] Reference is made to European Patent Publication Number 0981110, entitled A METHOD AND SYSTEM OF PRINTING A POSTAGE INDICIA FROM AN ENVELOPE DESIGN APPLICATION, assigned to the assignee of this application and filed on July 20, 1999.

[0004] Reference is made to European Patent Publication Number 0974914, entitled A METHOD AND SYSTEM FOR CAPTURING DESTINATION ADDRESSES FROM LABEL DATA, assigned to the assignee of this application and filed on July 20, 1999.

[0005] The present invention relates generally to the field of print stream data processing; and, more specifically, to the field of printing to multiple device drivers via a single print stream input.

[0006] Mail preparation systems, such as the DOCUMATCH™ mail processing and finishing system available from Pitney Bowes Inc. of Stamford, Connecticut, establish a mail piece print run at a host personal computer (PC) and then direct the stream to printer peripherals for printing to an envelope and/or to a page substrate. Mail preparation systems are an example of systems whose purpose is to utilize address lists, perform addressing hygiene through the use of address correction techniques, assign barcoding and, download data to addressing printers, collators, sealers, and the like, for the purpose of producing a mailpiece. These systems sometimes have only a single input/output (I/O) port interface between the PC and the document printer. Thus, the current mail preparation systems are generally constrained by their printer hardware architecture.

[0007] To support such a system architecture, the application print data stream must be altered to allow the generation of envelope print data streams and its re-injection into the main application print stream. The creation of the envelope print stream involves the capture of text data contained in the document to generate the envelope, the use of an envelope definition module, and the use of proprietary print protocol language for the mail preparation system to direct the data to an appropriate printer.

[0008] The print stream created by the main application is generally in the form of text data, though it may take on other forms. The data must be parsed and checked before format correction and barcoding techniques can be directed to the addresses in the text for creation of a mailpiece.

[0009] Mailpiece production systems are known in the art and have developed with changes in postal, service regulations (such as those of the United States Postal Service, or USPS) and with the proliferation of appropriate software applications. In turn, this production has served the need to automate and accelerate to accommodate growth.

[0010] As the United States Postal Service (USPS), together with the postal services of other countries around the world, moves toward more fully automated mail handling in an effort to contain costs while processing ever increasing volumes of mail, automated equipment which sorts and processes mail on the basis of machine readable postal codes, such as the "zip code" or other forms of postal coding, play an ever more significant role. In the United States, postal service regulations provide for a "Postnet" bar code which represents the five, nine, or eleven digit zip code of the destination address in a machine readable form. 4-State can be utilized within Canada.

[0011] Additionally, a system for printing envelopes with addresses including bar code is disclosed in commonly assigned U.S. Patent No. 5,175,691 for a SYSTEM AND METHOD FOR CONTROLLING AN APPARATUS TO PRODUCE ITEMS IN SELECTED CONFIGURATIONS; issued on December 29, 1992 to Baker et al, (hereinafter referred to as Baker), which describes a system for printing mail pieces which includes a printer for printing sheets and envelope forms and a folder-sealer mechanism for folding the envelope form around the sheets to form a mail piece, and a computer based control system for controlling the printer and folder. In the system of this application, when an operator is creating a file of letters to be printed, the operator may designate a selected field within each letter as containing the destination address. The system will then extract the information in this designated field and with it create a new page of material to be printed on the envelope form; and, if the address within the designated field includes a zip code, the system will add a corresponding barcode to the new page. The system then adds this new page to the file before the file is output.

[0012] The ability to structure software coding is extremely important when linking data to be downloaded to a printer being utilized in the addressing environment. U.S. Patent No. 5,583,970 for a PRINTER COMMAND SET FOR CONTROLLING ADDRESS AND POSTAL CODE PRINTING FUNCTIONS, issued December 10, 1996 to Strobel (hereinafter referred to as Strobel), and assigned to the assignee of the present claimed invention, is instructive in this respect.

[0013] Strobel is a method and system for printing images to a substrate wherein the commands normally input by an operator, or resident within the printer, can be determined at a host data processor. The system can control address and postal code printing functions beginning at the host computer together. The system will derive printing data, including address data, from a se-

lected application resident in the host computer. The host computer creates and then transmits printer command sets and printing data, via transmitting means to a microprocessor within the printer. The microprocessor drives a language interpreter which directs the printer commands to a parsing step for determining the address location from within the data to be printed. The language interpreter then assigns delivery point digits to a zip code that was isolated from the transmitted address data. The newly created zip code is then matched with the bar code data stored within the microprocessor's corresponding memory. A bar code corresponding to the new zip code is selected. The language interpreter then directs the printer's controller to prepare to print the address with its corresponding zip code, any graphics images that may have been included within the print data, and text, if any. The printer controller positions the bar code for printing, and then prints the bar code and address data, zip code, and any graphics images and text to an envelope or other substrate.

[0014] A particular limitation to current methods and systems, however, is found in the assembly of the envelope print stream which fuels the prior art detailed above. Mailpiece production systems having two separate printers and a single I/O port are constrained by the serial connection between their printers, making it impossible for the document printer to query information directly from its corresponding envelope printer. Therefore, as is illustrated by FIG. 2 hereinbelow, these mailpiece preparation systems must integrate two drivers while exposing only one to the application.

[0015] Therefore, it is an object of the present invention to provide for a method and system for determining and extracting an address from a print stream. Additionally, it is an object of the present invention to generate new print streams by printing to a secondary driver, reading the secondary print streams and injecting the secondary print stream back into the primary application print stream associated with a single I/O port connecting the mail preparation system to its host PC.

[0016] The invention is a method and system of modifying print stream data in a printing system having at least two printers and a single input/output port and comprises a number of steps and components.

[0017] The method begins by sending a print stream from a data processing application through a graphical device interface (GDI) to a print spooler to form a GDI print stream. The GDI print stream may contain: control data with a corresponding control page wizard which is utilized to facilitate mail merge functionality within the printing system; text data; address data; and/or, some other components. The data processing application can be a mailpiece designer application for preparing a mailpiece based on assigned parameters. Additionally, the mailpiece designer application is capable of presenting a data entry screen to a system user for performing the further steps of creating and/or modifying a mailpiece definition file and, storing and/or retrieving one or more

mailpiece definition files wherein each of the files corresponds to a specific mail print run. In a preferred embodiment, the document designer application is a 32-bit WINDOWS automation server.

5 [0018] The printing system employs a print stream monitor within a document driver kernel context for scanning the GDI print stream to determine whether or not the print stream comprises a set of text data and/or a set of address data. If the print stream comprises text data then the text data is tagged and sent to a user mode module; however, if the print stream does not comprise text data, then the print stream is sent directly to a data injection step. After tagging, the text data is stored in a local buffer. The tagged text stored in the local buffer cannot be retrieved until the stored tagged text has received an end of page control mark for the text sought to be retrieved.

10 [0019] The tagged and stored text data is then retrieved from the local buffer and it is determined as to whether or not an address is contained within the tagged text. The determination is made by an envelope parser for detecting, parsing, and then extracting address data from the print stream. If an address is found in the tagged text, then the address is placed in an envelope print format to create an envelope data set; however, if an address is not found, then the tagged text is sent directly to the data injection step. An envelope printer device context is then created and the envelope data set is transmitted to an envelope kernel for creating an envelope printer device language file.

15 [0020] The GDI print stream is converted by a document printer command language (PCL) generator into an envelope printer language. The envelope data resulting is then utilized by a second designer application for displaying a set of data fields of the envelope data to a system user, reading a set of parameters created by the second designer application; and, writing the envelope data to a printer driver. The envelope data set is then printed.

20 [0021] Upon printing the envelope data set, the printer device language is then read by the print stream monitor which is used to modify the print stream by taking the envelope data set and injecting it back into the print stream from which it was extracted by merging the set of text data and the set of envelope data. The print stream is then transmitted to a next destination such as a document printer where a control page parser for detecting, parsing, and then extracting the document data from the print stream is employed. A printer command language (PCL) generator for converting the print stream into a document printer language is then employed. A printer driver is then activated for causing a printer to print the document data to one or more sheets.

25 [0022] FIG. 1 is a block diagram of a system within which the method of the present invention could reside and be utilized.

30 [0023] FIG. 2 is a flowchart of the prior art method of printing envelope data extracted from a print stream in

a WINDOWS® 95 environment.

**[0024]** FIG. 3A is a flowchart of the method of the present invention showing the initiation of the print stream and including a document driver kernel.

**[0025]** FIG. 3B is a continuation of the flowchart of the method of the present invention beginning with the document driver user mode module and concluding with the re-injection of data into the print stream.

**[0026]** FIG. 4 is a block diagram of the system of the present invention.

**[0027]** The system concept and architecture describes the method and system required to print a mail piece run as defined by a user from a WINDOWS® application document and/or an address list to system printer. The printing application can be, but is not limited to, a word processor application and a mail list management or database application. Consequently, the system requirements include: the interception of the print stream data from the documents being printed; parsing and extraction of certain information from the print stream data; adding information into the print stream data to generate a finished mail piece from the document being printed; and, converting the graphical device interface (GDI) print stream into a printer language print stream.

**[0028]** The following detailed description of the preferred embodiments presents the system requirements in context,

**[0029]** Turning to FIG. 1, there is shown a block diagram of a system within which the method of the present invention could reside and be utilized.

**[0030]** System 10 comprises a microprocessor 12 interoperatively connected to monitor 14 for viewing the representation of the medium (such as an envelope or label) to be acted upon by the design application 22. The viewing of the media representation on monitor 14 promotes ease of use in selecting the various options available to the system user while formatting the medium, and provides an example of the human interface that can be brought to system 10. The monitor 14, under control of the design application 22, is able to show the system user: the medium representation; available menus from which option selections may be made; the medium's indicia; the amount of postage that will be incorporated into the indicia; and varied print fields available for printing to the selected medium. Microprocessor 12 is interoperatively connected to scanner 16. Scanner 16 provides system 10 with the ability to scan address field data, barcodes, or other scannable data sources as an input to design application 22. Printer 26 is also interoperatively connected to microprocessor 12 and serves as the output device by which the print fields are printed to the selected medium. Additionally, keyboard 20 is interoperatively connected to microprocessor 12 and serves as an input device for the input of data. Modem 18 gives system 10 the ability to communicate with other systems via communications means of varied types or to download print fields for remote storage; and, memory 24 al-

lows the system to retain data for use in maintaining records or for storing data for future use.

**[0031]** Turning to FIG. 2, there is shown a flowchart of the prior art method of printing envelope data extracted from a print stream in a WINDOWS 95 environment.

**[0032]** The prior art method begins at step 100 where a data processing application such as a mailpiece preparation application, operating in a WINDOWS 95 environment, initiates a print stream for each printed document. From step 100, the method advances through a graphical device interface (GDI) at step 102 before entering a document driver module that begins at step 104.

**[0033]** At step 104, the method queries as to whether or not text data has been detected within the print stream. If the response to the query is "NO," then the method proceeds directly to step 118 where the data is re-injected into the print stream before advancing to step 120 where the sequence ends while the print stream is directed toward another peripheral device. If the response to the query at step 104 is "YES," however, then the method advances to step 106 where the text data is stored in a local buffer to await an "end-of-page" control mark from the system.

**[0034]** At step 108, the method queries as to whether or not an end-of-page control mark has been received at the local buffer. If the response to the query is "NO," then the method proceeds directly to step 118 where the data is re-injected into the print stream before advancing to step 120 where the sequence ends while the print stream is directed toward another peripheral device. If the response to the query at step 108 is "YES," however, then the method advances to step 110 where the data is retrieved from the local buffer before advancing to the query at step 112.

**[0035]** At step 112, the method queries as to whether or not an address has been found in the retrieved text. If the response to the query is "NO," then the method proceeds directly to step 118 where the data is re-injected into the print stream before advancing to step 120 where the sequence ends while the print stream is directed toward another peripheral device. If the response to the query at step 112 is "YES," however, then the method advances essentially simultaneously to steps 114 and 116. At step 114, the address data is printed to an envelope as envelope data, while at step 116, an escape sequence is created using built-in printer command language (PCL) commands. Steps 114 and 116 rejoin at step 118 where the data is re-injected into the print stream before advancing to step 120 where the sequence ends while the print stream is directed toward another peripheral device such as a document printer.

**[0036]** Next turning to FIG. 3A, there is shown a flowchart of the method of the present invention showing the initiation of the print stream and including a document driver kernel.

**[0037]** The method begins at step 200 where a data processing application such as a mailpiece preparation application, operating in a WINDOWS NT environment,

initiates a print stream for each printed document. From step 200, the method advances through a graphical device interface (GDI) at step 202 and sends the print stream to a print spooler at step 204 before entering document driver kernel 205 that begins with a query at step 206. The GDI enables the system printer driver by initiating the query,

**[0038]** At step 206, the method queries as to whether or not text data has been detected within the print stream. If the response to the query is "NO," then the method proceeds to step 208 where the data is sent to the user mode module, via path A to be re-injected into the print stream at step 236 as is shown in FIG. 3B. If the response to the query at step 206 is "YES," however, then the method advances to step 210 where the text data is tagged and sent to the document driver user mode module 225, along path B as is shown in FIG. 3B.

**[0039]** FIG. 3B is a continuation of the flowchart of the method of the present invention beginning with the document driver user mode module 225 and concluding with the re-injection of data into the print stream at step 236.

**[0040]** Turning first to step 212, path B is shown re-entering the method flow and the tagged text data is stored in a local buffer. From step 212, the method advances to the query at step 214. At step 214, the method queries as to whether or not an end-of-page control mark has been received at the local buffer. If the response to the query is "NO," then the method proceeds directly to step 236 where the data is re-injected into the print stream before advancing to step 238 where the sequence ends while the print stream is directed toward another peripheral device. If the response to the query at step 214 is "YES," however, then the method advances to step 216 where the data is retrieved from the local buffer before advancing to the query at step 218.

**[0041]** At step 218, the method queries as to whether or not an address has been found in the retrieved text. If the response to the query is "NO," then the method proceeds directly to step 236 where the data is re-injected into the print stream before advancing to step 238 where the sequence ends while the print stream is directed toward another peripheral device. If the response to the query at step 218 is "YES," however, then the method advances essentially simultaneously to step 220. At step 220, the system creates an envelope printer device context before advancing to step 222. At step 222, the address data is printed to an envelope driver as envelope data.

**[0042]** From step 222, the method advances essentially simultaneously to steps 224 and 228. Step 224 exists as a separate envelope kernel and user mode where an envelope printer device language (PDL) is generated for filing before advancing to step 226 where the envelope print data is placed in a temporary file. The method advances from step 226 to step 234 where the envelope PDL data file is read and then injected at step 236 back into the print stream before advancing to step 238 where

the sequence ends while the print stream is directed toward another peripheral device such as a document printer.

**[0043]** Turning back to step 228, the method queries as to whether or not all the available envelope data for this particular document has been printed. If the response to the query is "NO," then the method returns to re-enter the method flow at step 222 so that the remaining data can be printed. If the response to the query at step 228 is "YES," however, then the method advances to step 230 where the envelope printing is ended. The method then advances to a query at step 232. At step 232, the method queries as to whether or not the system is ready to inject the envelope data back into the print stream. If the response to the query is "NO," then the method returns to re-enter the method flow at step 232 and the query is repeated. If the response to the query at step 232 is "YES," however, then the method advances to step 234 where the envelope PDL data file is read and then injected at step 236 back into the print stream before advancing to step 238 where the sequence ends while the print stream is directed toward another peripheral device such as a document printer.

**[0044]** FIG. 4 is a block diagram of the system and corresponding components of the present invention.

**[0045]** A microprocessor 310 is shown interoperatively connected to a document design application 312, for preparing a document such as a mailpiece with its associated text insert, to be printed as document text and as envelope text. In a preferred embodiment of the present invention, the document designer application 312 is a 32-bit WINDOWS automation server. The document designer application is capable of creating and/or modifying a mailpiece definition file and storing and/or retrieving one or more mailpiece definition files wherein each of said files corresponds to a specific mail print run and results in a print stream. Also connected to microprocessor 310, is an envelope design application 314.

**[0046]** The envelope design application 314 is utilized for: displaying a set of data fields of the envelope text data portion to a system user; reading a set of parameters created by the envelope designer application; and, writing the envelope data to a printer driver. The set of data fields displayed is representative of the face of an envelope (comprising an indicia print field and an addressee print field), thus allowing the system user a convenient way to check on field selection and placement.

**[0047]** Additionally connected to microprocessor 310 is a print stream monitor 318 for: scanning the print stream generated by the mailpiece design application 312; detecting a set of document data or control data and a set of envelope data; interfacing with the envelope parser 320 to extract an address from the document text data; interfacing with the document parser 322 to extract control page information from the print stream; generating the envelope PCL print data at the PCL print generator 324; and, for modifying the print stream to merge

the two sets of data. The print stream monitor 318 maintains the system timing during printing of the mailpiece and the general performance of the document print job.

[0048] The system includes a document (or control page) parser 322 for detecting, parsing, and then extracting the document data from the print stream, as well as instructing the print stream task manager (not shown). Further included, is an envelope parser 320 for detecting, parsing, and then extracting the envelope data from the print stream and then indicating to the print stream task manager that an address has been detected.

[0049] To print the envelope, a PCL generator 324 is connected to the microprocessor 310 for converting the envelope data as extracted from the print stream into a second printer language, thus creating a proper PCL for the envelope text to be printed through printer driver 328 to printer 332 and on to an envelope or similar substrate. To print the document, a PCL generator 326 is connected to the microprocessor 310 for converting the document data as extracted from the print stream into a second printer language, thus creating a proper PCL for the document text to be printed through printer driver 330 to printer 334 and on to one or more sheets or similar substrate. The WINDOWS NT printer architecture requires that every printer driver be implemented as a pair of user mode dynamic link libraries (DLL), as well as a printer specific component.

[0050] The WINDOWS NT printing architecture is part of the NT graphics architecture and consists of three components; these are: the applications that interface with the WINDOWS GDI; the server print spooler that interfaces with the print services convention; and, the kernel mode print services that include the printer driver minidriver and the I/O port interface. The print spooler internally accesses the user interface and the user mode printer driver components. The minidriver is a data file that contains the printer data tables as well as code specific to system driver that works in conjunction with the shim common driver. The purpose of the shim common driver is to intercept the raster graphic entry points to obtain print stream data and to provide each of the other drivers with common kernel functions when necessary.

[0051] While certain embodiments have been described above in terms of the system within which the method may reside, the invention is not limited to such a context.

[0052] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

## Claims

1. A method of modifying print stream data in a printing system, said method comprising the steps of:

(a) sending a print stream from a data processing application to a print spooler;  
(b) determining, in a document driver kernel, whether or not said print stream comprises text data, and;

(i) if said print stream comprises text data then tagging said text data and sending said tagged text data to a user mode module; or  
(ii) if said print stream does not comprise text data then sending said print stream directly to a data injection step;

(c) storing said tagged text in a local buffer;  
(d) retrieving said tagged text from said local buffer and determining whether or not an address is contained within said tagged text, and:

(i) if an address is found in said tagged text, then placing said address in an envelope print format to create an envelope data set; and  
(ii) if an address is not found then sending said tagged text directly to said data injection step;

(e) creating an envelope printer device context and transmitting said envelope data set to an envelope kernel for creating an envelope printer device language file and then printing said envelope data set;

(f) reading said printer device language and then injecting said envelope data set into said print stream; and

(g) transmitting said print stream to a next destination.

2. The method of claim 1, wherein said print stream is passed through a graphical device interface (GDI) when being sent from said data processing application to said print spooler to form a GDI print stream.

3. The method of claim 1, wherein said print stream comprises control data.

4. The method of claim 1, wherein said local buffer stores said tagged text until at least one end-of-page control mark is received in said local buffer.

5. The method of claim 1, wherein said tagged text stored in said local buffer cannot be retrieved until said stored tagged text has received an end of page

control mark for said stored tagged text sought to be retrieved.

6. The method of claim 1, wherein said data processing application is a mailpiece designer application. 5
7. The method of claim 6, wherein said mailpiece designer application is capable of presenting a data entry screen to a system user for performing the further steps of: 10
  - (a) creating and/or modifying a mailpiece definition file; and
  - (b) storing and/or retrieving one or more mailpiece definition files wherein each of said files corresponds to a specific mail print run. 15
8. The method of claim 1, wherein said print stream comprises a control page wizard. 20
9. The method of claim 8, wherein said control page wizard is utilized to facilitate mail merge functionality within said printing system.
10. The method of claim 2, wherein said GDI print stream is converted by a document printer command language (PCL) generator into an envelope printer language, 25
11. A document printing system having at least two printers and a single input/output port, said system comprising; 30
  - (a) a first document designer application for preparing a document to be printed in at least a first portion and a second portion and wherein said preparation results in a print stream; 35
  - (b) a second designer application for:
    - (i) displaying a set of data fields of said second portion to a system user; 40
    - (ii) reading a set of parameters created by said second designer application; and
    - (iii) writing said second portion to a printer driver; 45
  - (c) a print stream monitor for:
    - (i) scanning said print stream to detect a set of first portion data and a set of second portion data; and 50
    - (ii) modifying said print stream to merge said set of first portion data and said set of second portion data;
    - (d) a control page parser for detecting, parsing, and then extracting said first portion data from said print stream; 55

(e) a second portion parser for detecting, parsing, and then extracting said second portion data from said print stream;  
 (f) a first printer command language (PCL) generator for converting said print stream into a first portion printer language;  
 (g) a second (PCL) generator for converting said second portion as extracted from said print stream into a second printer language;  
 (h) a first printer driver for causing said first printer to print, utilizing said first portion printer command language, said first portion data to one or more sheets; and  
 (i) a second printer driver for causing said second printer to print, utilizing said second printer command language, said second portion data to a substrate.

12. The document printing system of claim 11, wherein said first document designer application is a 32-bit WINDOWS automation server.
13. The document printing system of claim 12, wherein said first document designer application is capable of creating and/or modifying a mailpiece definition file and storing and/or retrieving one or more mailpiece definition files wherein each of said files corresponds to a specific mail print run.
14. The document printing system of claim 11, wherein said document designer application is for producing one or more mailpieces and wherein said first portion comprises document data and said second portion comprises envelope data.
15. The document printing system of claim 11, wherein said set of data fields is representative of the face of an envelope.
16. The document printing system of claim 15, wherein said set of data fields further comprises an address-ee print field.
17. The document printing system of claim 15, wherein said set of data fields further comprises an indicia print field.
18. The document printing system of claim 11, wherein said second portion data is converted by a document printer command language (PCL) generator into an envelope printer language.
19. A mailpiece creation system having a first printer and a second printer and a single input/output port, said system comprising:
  - (a) a host computer having a mailpiece designer application for generating a print stream and

wherein said print stream is representative of said mailpiece;

(b) an envelope designer application for:

- (i) displaying a set of envelope data fields to a system user; 5
- (ii) reading a set of envelope parameters created by said envelope designer application; and
- (iii) writing said envelope to a printer driver; 10

(c) a print stream monitor for:

- (i) scanning said print stream to detect address data resident in said print stream and control page or control mark data resident in said print stream; and 15
- (ii) modifying said print stream to merge a set of document page data and a set of address data; 20

(d) a control page parser for detecting, parsing, and then extracting said document page data from said print stream;

(e) an addressing parser for detecting, parsing, and then extracting address data from said print stream; 25

(f) a document printer command language (PCL) generator for converting said print stream into a document printer language; 30

(g) an envelope (PCL) generator for converting text addresses extracted from said print stream into an envelope printer language;

(h) a first printer driver for causing said first printer to print said address data to an envelope utilizing said envelope printer language; and 35

(i) a second printer driver for causing said second printer to print said document data to one or more sheets utilizing said document printer language. 40

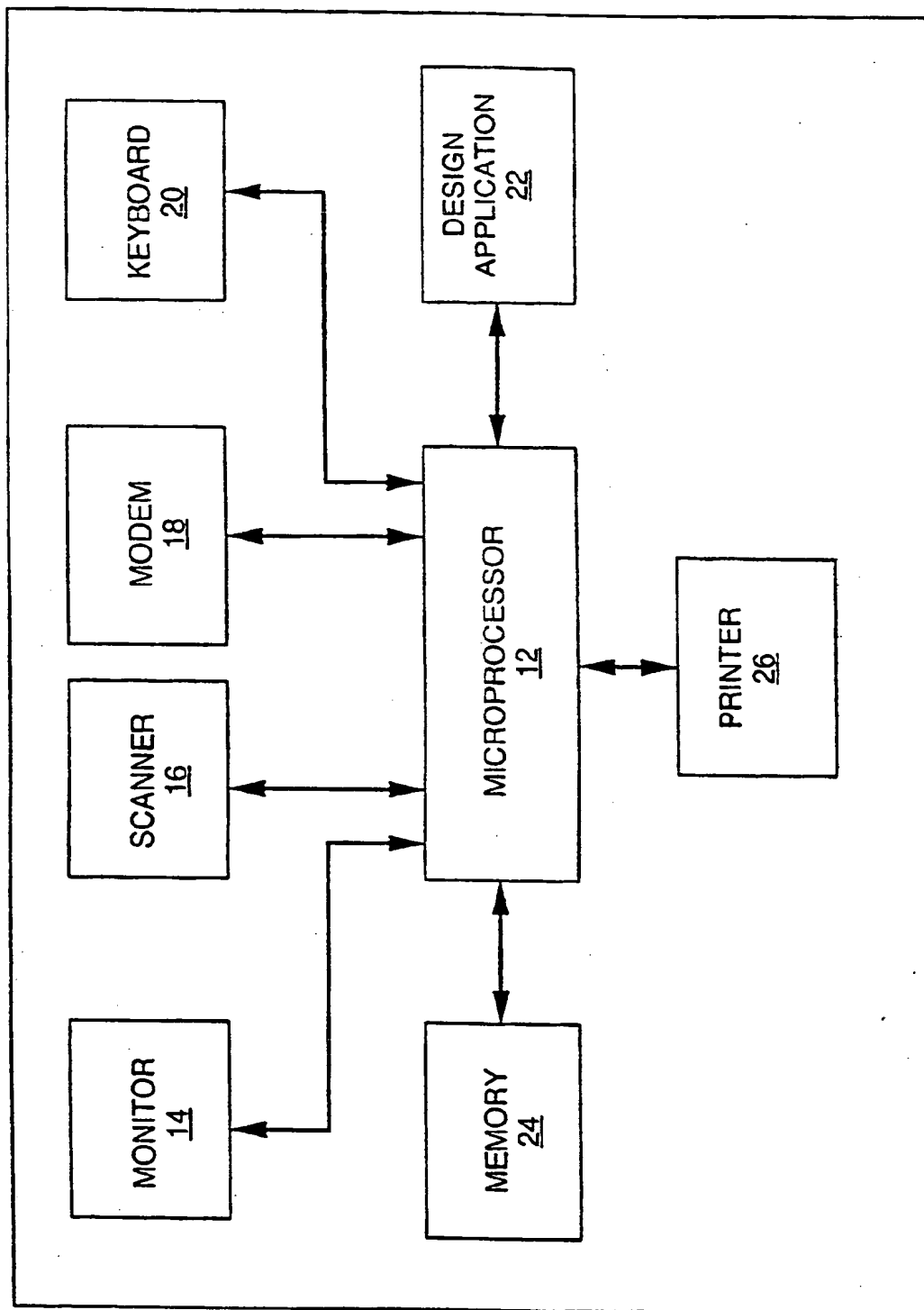
20. The mailpiece creation system of claim 19, wherein an envelope printer device context is established to allow said address data to be printed to said second printer driver for printing to said envelope. 45

50

55



FIG. 1



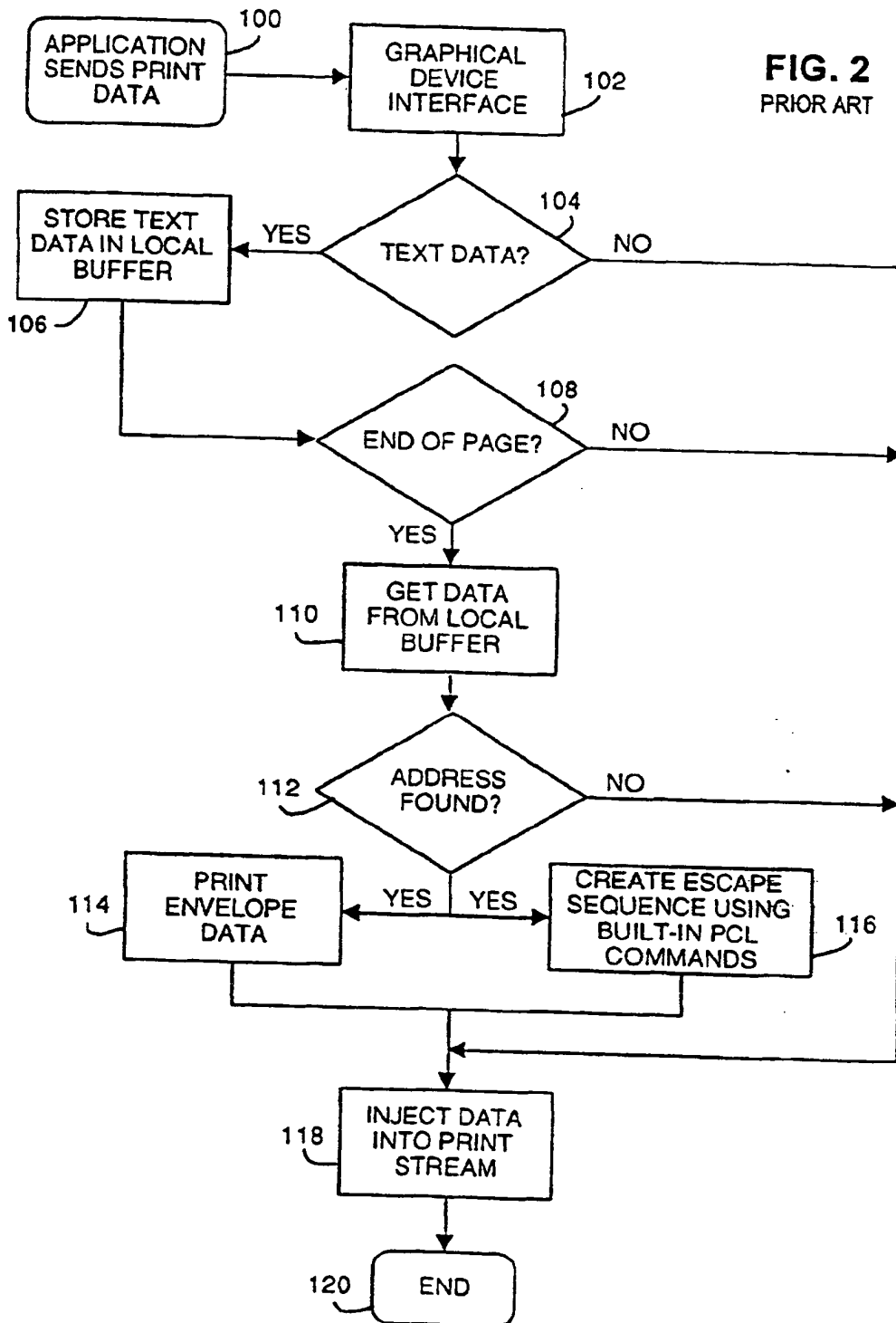
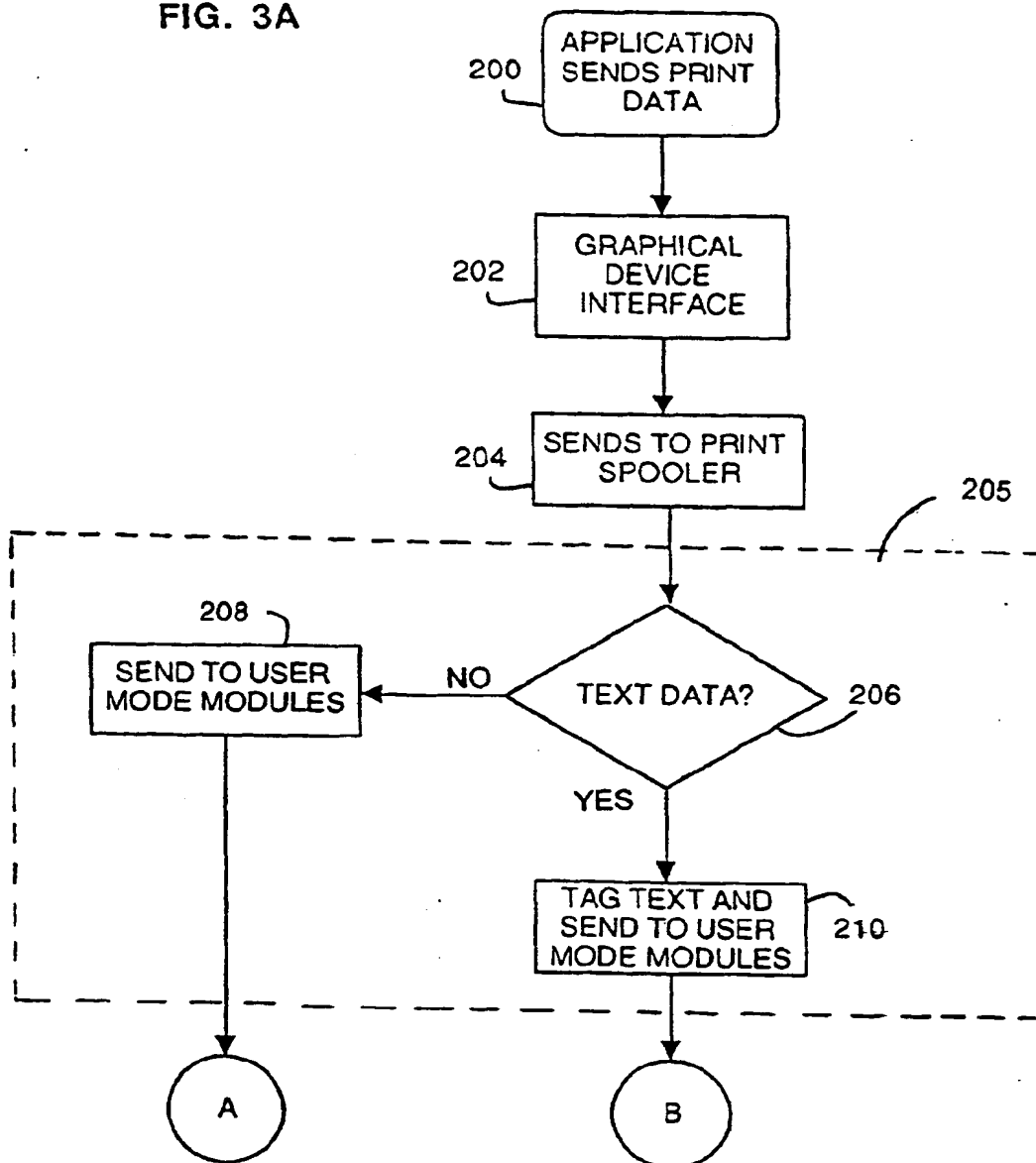
**FIG. 2**  
PRIOR ART

FIG. 3A



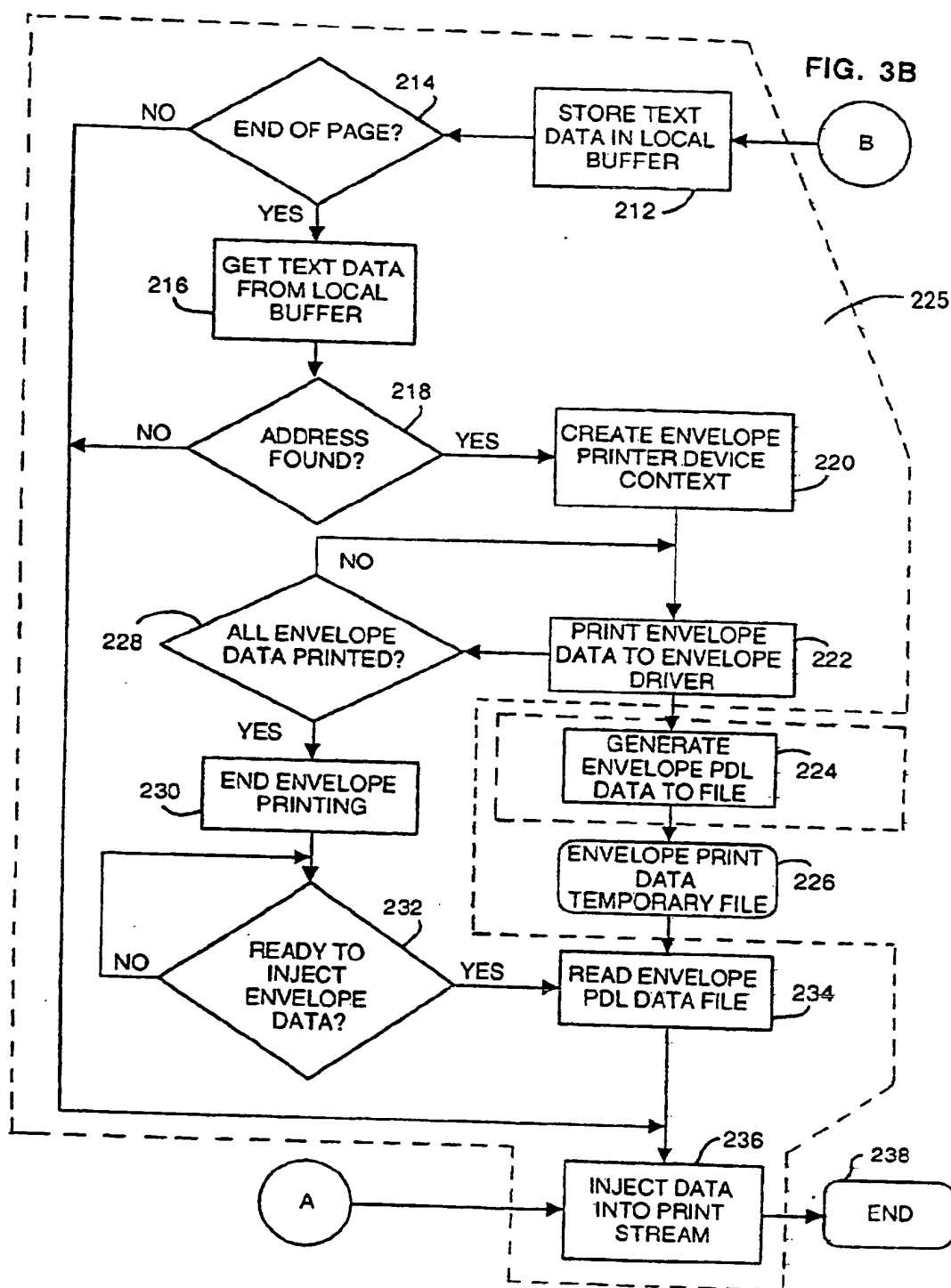
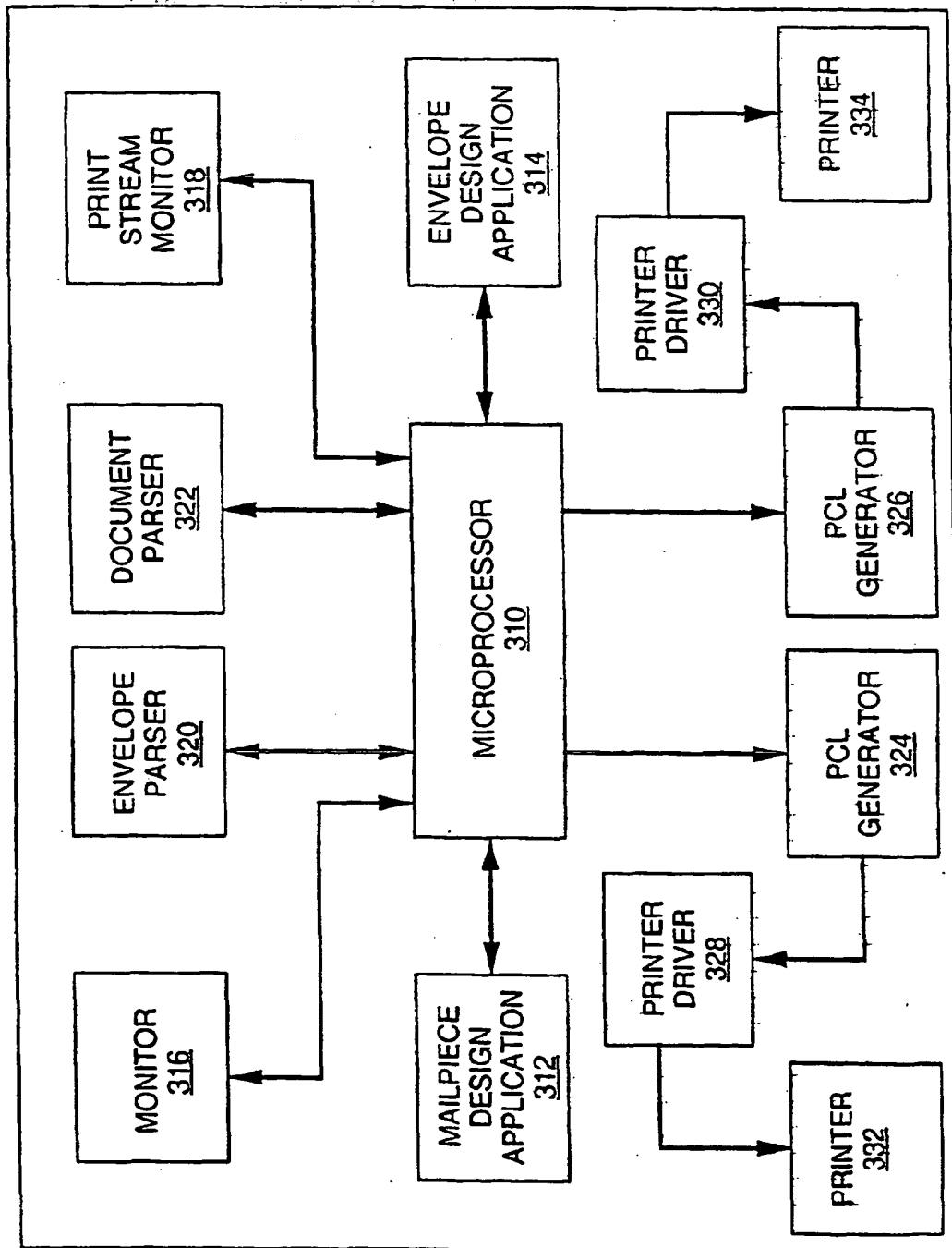
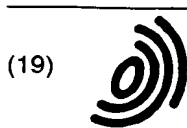


FIG. 4



**THIS PAGE BLANK (USPTO)**



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11) **EP 1 139 275 A3**

(12) **EUROPEAN PATENT APPLICATION**

(88) Date of publication A3:  
14.01.2004 Bulletin 2004/03

(51) Int Cl.7: **G06K 15/00, G06F 17/27,  
G07B 17/00**

(43) Date of publication A2:  
04.10.2001 Bulletin 2001/40

(21) Application number: **01107742.7**

(22) Date of filing: **02.04.2001**

(84) Designated Contracting States:  
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU  
MC NL PT SE TR**  
Designated Extension States:  
**AL LT LV MK RO SI**

- **Clemens, Daniel J.**  
Deer Harbor, Washington 98243 (US)
- **Girardi, Victor**  
Oxford, Connecticut 06478-1839 (US)
- **Rolling, Patrick**  
Cheshire, Connecticut 06410 (US)

(30) Priority: **31.03.2000 US 541001**

(71) Applicant: **PITNEY BOWES INC.**  
Stamford, Connecticut 06926-0700 (US)

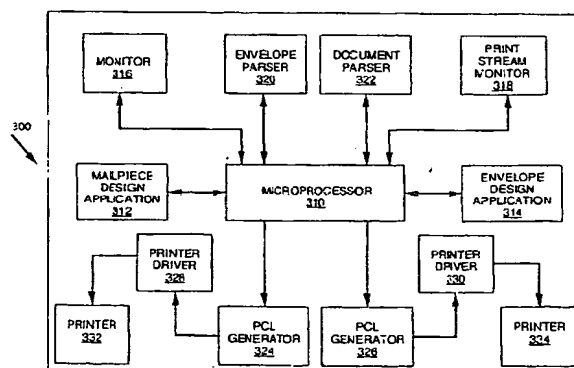
(74) Representative: **HOFFMANN - EITLE**  
Patent- und Rechtsanwälte  
Arabellastrasse 4  
81925 München (DE)

(72) Inventors:  
• **Bratsamos, James S.**  
Seattle, Washington 98122 (US)

(54) **A method and system of print stream address extraction**

(57) The invention is a method and system of modifying print stream data. The method begins by sending a print stream from an application through a graphical device interface to a print spooler. A print stream monitor determines whether or not the print stream comprises a set of text data and/or a set of address data. If the print stream comprises text data, then the text data is tagged and sent to a user mode module where it is stored in a local buffer. The tagged and stored text data is then retrieved and parsed for any address which may be contained within the tagged text. The address data is then extracted from the print stream and placed in an envelope print format to create an envelope data set. An envelope printer device context is then created and the envelope data set is transmitted to an envelope kernel for creating an envelope printer device language file. The print stream is converted by a document printer command language (PCL) generator into an envelope printer language. The envelope data set is then printed. The envelope data set is injected back into the print stream from which it was extracted by merging the set of text data and the set of envelope data. The print stream is then transmitted to a next destination such as a document printer, where a printer driver is then activated for causing a printer to print the document data to one or more sheets.

FIG. 4



EP 1 139 275 A3



European Patent  
Office

# EUROPEAN SEARCH REPORT

Application Number  
EP 01 10 7742

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	EP 0 927 581 A (PITNEY BOWES) 7 July 1999 (1999-07-07) * abstract * * column 5, line 40 - column 10, line 10 * * figures 2,3 *	1-4,6,7, 10-20	G06K15/00G06F17/2 G06F17/21 G07B17/00 G06K15/00
D,A	EP 0 974 915 A (PITNEY BOWES) 26 January 2000 (2000-01-26) * abstract * * column 7, line 52 - column 15, line 3 * * figures 6,7 *	1-4,6,7, 10-20	
A	US 5 650 934 A (MANDULEY FLAVIO M) 22 July 1997 (1997-07-22) * abstract * * column 3, line 26 - column 5, line 62 * * figures 1-4 *	1-4,6,7, 10-20	
			TECHNICAL FIELDS SEARCHED (Int.Cl.7)
			G06F G07B G06K
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 21 November 2003	Examiner Jacobs, P
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons &amp; : member of the same patent family, corresponding document</p>			

EPO FORM 1503 33 82 (P04C01)



**ANNEX TO THE EUROPEAN SEARCH REPORT  
ON EUROPEAN PATENT APPLICATION NO.**

EP 01 10 7742

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on  
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

21-11-2003

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
EP 0927581	A	07-07-1999	US	6320670 B1	20-11-2001
			DE	69815043 D1	03-07-2003
			EP	0927581 A2	07-07-1999
-----					
EP 0974915	A	26-01-2000	US	6337743 B1	08-01-2002
			EP	0974915 A2	26-01-2000
-----					
US 5650934	A	22-07-1997	CA	2177447 A1	01-12-1996
			EP	0745435 A2	04-12-1996
-----					

EPC FORM P459

For more details about this annex: see Official Journal of the European Patent Office, No. 12/82

**THIS PAGE BLANK (USPTO)**